

Title	一般化ブロックパズルのPSPACE完全性の別証明 (計算機科学基礎理論の新展開)
Author(s)	北川, 智博; 岩田, 茂樹
Citation	数理解析研究所講究録 (2003), 1325: 209-214
Issue Date	2003-05
URL	http://hdl.handle.net/2433/43199
Right	
Type	Departmental Bulletin Paper
Textversion	publisher

一般化ブロックパズルの PSPACE 完全性の別証明

北川 智博(Tomohiro Kitagawa), 岩田 茂樹(Shigeki Iwata)

電気通信大学 情報工学科

盤面上に配置された駒を縦または横に平行移動して、特定の駒を特定の位置に移動させるパズルをブロックパズルと呼ぶ。ブロックパズルは、欧米では L'Ane Rouge の名で[1]、日本では「箱入娘」の名で知られている。ブロックパズルの盤面の大きさを $n \times n$ に拡張し、拡張した盤面上でのブロックパズルが解けるかどうかの一般化ブロックパズル問題の複雑性が PSPACE 完全であることはすでに[2]で証明されているが、本発表では別証明を与える。この研究の背景には Mine-Sweeper の NP 完全性[3]、倉庫番の PSPACE 完全性[4]、一般化詰将棋の指数時間完全性[5]などパズル問題の複雑性が示されている。

1. 最初に

ブロックパズルにはいろいろな種類が存在するが[1]、ここでは「箱入娘」を例に挙げ、そこから一般化ブロックパズルを定義する。「箱入娘」は縦5、横4からなる盤面の上に(5×4と表記する)、1×1、1×2、2×2の駒が図1のように配置されている。その中で各駒を縦または横に平行移動させていき、最終的に「娘」の駒を盤上の斜線部分にあたる出口から盤の外部に移動できるとき「箱入娘」は解ける。なお「娘」以外の駒は出口から盤の外部に移動できない。

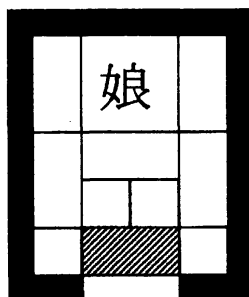


図1 箱入娘

このようなパズルを盤面の大きさ $n \times n$ に、駒の大きさ $p \times q$ (p, q は6以下の整数とする) に拡張したものを一般化ブロックパズル(以下 GBP)と呼ぶ。また GBP には「娘」に対応する駒と盤上に出口が与えられる。ある GBP が与えられたとき、「娘」に対応する駒が出口に移動できるかどうか判定する問題を GBP 問題と定義する。GBP の形式的な定義は次節で説明する。GBP 問題の PSPACE 完全性につ

いては Hearn 他により 2002 年の ICALP[2]で示された。ここでは別証明を与える。他のパズル問題の複雑性としては Mine-Sweeper の NP 完全性、倉庫番の PSPACE 完全性、一般化詰将棋の指数時間完全性などが示されている。

GBP 問題が PSPACE 完全であることを示すためには以下のような段階を経る。

- I. pebble game[6](以下 PG) に双方向性を持たせた **symmetric pebble game**(以下 SPG) を定義し、**symmetric Turing machine**(以下 STM)[7]の入力受理問題が **SPG 問題**に多項式時間還元可能であることを示すことによって SPG 問題が PSPACE 完全であることを示す。
- II. SPG を変形する。
- III. II で変形した SPG を利用して SPG 問題が GBP 問題に多項式時間還元可能であることを示し、GBP 問題が PSPACE 完全であることを示す。

PG, SPG, SPG 問題, STM の定義は次節で説明する。

2. 諸定義

GBP は5組のゲーム $G = (X, F, p, S, t)$ である。

X は $n \times n$ (n は自然数)マス上のブロックパズルの盤面の集合である。

F は駒の有限集合である。駒は互いに重ならないよう縦または横に平行移動できる。また駒の大きさは $i \times j$, $1 \leq i \leq 6$, $1 \leq j \leq 6$ である。

$p \in F$ は特定の駒で「娘」に相当する。

$S \in X$ は盤の初期配置である。

t は盤上の特定の位置であり出口と呼ぶ。

$A, B \subseteq X$ で駒を平行移動して盤面が A から B に変わったとき $A \rightarrow B$ と表す。もし $S \rightarrow A$, A の中の t には p が配置されているならばこのゲームは解けるという。ただし、 \rightarrow は \rightarrow の反射推移閉包である。

PG[6]は4組のゲーム $G = (X, R, S, T)$ である。

X は頂点の有限集合である。

$R \subseteq \{x, y, z : x, y, z \in X, x \neq y, y \neq z, z \neq x\}$ はルールの有限集合であり、 $A, B \subseteq X$, $(x, y, z) \in R$, $x, y \in A$, $z \notin A$, $B = (A - \{x\}) \cup \{z\}$ のときル

ール (x, y, z) で A から B に移動したと言い、 $A \longrightarrow B$ と表す。

$S \subseteq X$ は初期配置を表す。

$T \subseteq X$ は出口を表す。もし $S \xrightarrow{*} A$, $A \cap T \neq \emptyset$, になるような $A \subseteq X$ が存在するなら、このゲームは解けるという。ただし、 $\xrightarrow{*}$ は \longrightarrow の反射推移閉包である。

ある PG が与えられたとき初期配置からルールに従って石を動かしていき、石が出口に到達できるかを判定する問題を PG 問題という。

SPG は以下を満たす PG (X, R, S, T) のことである。

$$(x, y, z) \in R \text{ ならば } (z, y, x) \in R$$

以降 SPG で $(x, y, z) \in R$ と書いたときは $(z, y, x) \in R$ も含まれているものとする。また $(x, y, z) \in R$ を図 2 のように表す。頂点 x, y, z に対し、 x, z 間を無向辺 xz で結ぶ。さらに頂点 y から辺 xz に有効辺 y_x を結ぶ。

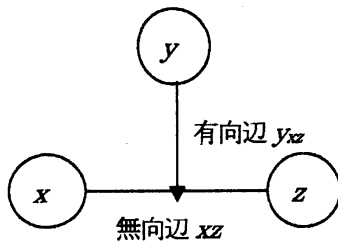


図 2 ルール (x, y, z)

以下 SPG の頂点とルールを図で示す場合は図 2 のように表現する。ある SPG が与えられたとき初期配置からルールに従って石を動かしていき、石が出口に到達できるかを判定する問題を SPG 問題という。

次に STM [7] $M = (Q, \Sigma, \Gamma, \delta, q_0, q_i)$ を定める。

Q は状態の有限集合である。

Σ はテープ記号の有限集合である。

$\Gamma \subseteq \Sigma$ は入力記号の集合である。

δ は状態遷移関数である。

$q_0 \in Q$ は初期状態である。

$q_i \in Q$ は受理状態である。

STM が普通の Turing machine (以下 TM) と違うところは状態遷移関数 δ が

$$\begin{cases} \delta(p, ab) \longrightarrow (q, cd, D) \\ \delta(q, cd) \longrightarrow (q, ab, -D) \end{cases}$$

$$p, q \in Q \quad a, b, c, d \in \Sigma \quad D \in \{0, 1\}$$

のように 1 対で定まっている。このとき $(p, (ab, D, cd), q)$ の形で記述することにする。また STM はテープヘッドの直前、直後のテープ記号を読むことができる。

$D=1$ のとき、すなわち $(p, (ab, 1, cd), q)$ は (A) または (B) の動作を行う。

(A) 状態が p で、かつテープヘッドが指すセルの記号が a で、かつテープヘッドの右隣にあるセルの記号が b であるならば次を実行する。 a を c に書き換える。テープヘッドを右に進める。 b を d に書き換える。状態を p から q に移す。

(B) 状態が q で、かつテープヘッドが指すセルの記号が c で、かつテープヘッドの左隣にあるセルの記号が d であるならば次を実行する。 c を a に書き換える。テープヘッドを左に進める。 d を b に書き換える。状態を q から p に移す。

$D=0$ のときはテープヘッドが元の位置に戻る点を除いて $D=1$ の場合と同様である。

$S(n)$ 領域限定決定性 TM が受理する問題のクラスを $DSPACE(S(n))$ 、 $S(n)$ 領域限定 STM が受理する問題のクラスを $SSPACE(S(n))$ 、 $S(n)$ 領域限定非決定性 TM が受理する問題を $NSPACE(S(n))$ とすると

$$\begin{aligned} DSPACE(S(n)) &\subseteq SSPACE(S(n)) \\ &\subseteq NSPACE(S(n)) \end{aligned}$$

という関係が知られている [7]。よって

$$\begin{aligned} DSPACE(poly) &= SSPACE(poly) \\ &= NSPACE(poly) \\ &= PSPACE \end{aligned}$$

が成り立つ。

3. SPG 問題の PSPACE 完全性

定理 1. SPG 問題は PSPACE 完全である。

(証明) PG 問題はすでに PSPACE 完全であることが知られている [6]。よって SPG 問題も PSPACE で解ける。

SPG 問題が PSPACE 困難であることを示すには、与えられた多項式領域限定 STM M と入力 x に対し、

M が x を受理する \Leftrightarrow SPG が解ける

を満たす SPG を多項式時間で構成すればよい。STM を $M = (Q, \Sigma, \Gamma, \delta, q_0, q_i)$ とする。また n を入力の長さとする。SPG $G = (X, R, S, \{t\})$ は以下のように構成する。

X の頂点として M の状態に対応する頂点、テープの各セルの記号に対応する頂点、およびテープヘッドに対応する頂点と状態遷移関数の動作を表す頂点を用意する。状態に対応する頂点は $[p]$, $p \in Q$ で、 M の状態が p の場合 $[p]$ に石を置く。 i 番目のセルに対応する頂点は $[i, a]$, $a \in \Sigma$ で、 M の i 番目のセル

のテープ記号が a の場合 $[i, a]$ に石を置く。ただし i は $1 \leq i \leq n'$, l は定数である。テープヘッドに対応する頂点は (i) で、 M のテープヘッドが (i) 番目のセルにある場合 (i) に石を置く。さらに状態遷移関数の動作を表す補助的な頂点として $[i, j, k]$, $1 \leq k \leq 5$ を用意する。これは i 番目のセルで δ の j 番目の動作を表す。

S の要素は、初期状態に対応する頂点と、テープヘッドの初期位置に対応する頂点である (1) と、各テープセルの入力記号列に対応する頂点とする。

i は M の受理状態に対応する頂点とする。

M が状態 p から状態 q に移る j 番目の状態遷移である $(p, (ab, D, cd), q)$ を持つとき、 $1 \leq i \leq n'$ に対し R は次の(1)から(9)を含む。

- (1) 状態が p であることを確認するルール $((i), [p], [i, j, 1])$ 。
- (2) i 番目のセルのテープ記号が a であることを確認するルール $([i, j, 1], [a, i], [i, j, 2])$ 。
- (3) $i+1$ 番目のセルのテープ記号が b であることを確認するルール $([i, j, 2], [b, i+1], [i, j, 3])$ 。
- (4) 状態を p から q に変えるルール $([p], [i, j, 3], [q])$ 。
- (5) i 番目のセルのテープ記号を a から c に変えるルール $([a, i], [i, j, 3], [c, i])$ 。
- (6) $i+1$ 番目のセルのテープ記号が b から d に変えるルール $([b, i+1], [i, j, 3], [d, i+1])$ 。
- (7) $i+1$ 番目のセルのテープ記号が d であることを確認するルール $([i, j, 3], [d, i+1], [i, j, 4])$ 。
- (8) i 番目のセルのテープ記号が c であることを確認するルール $([i, j, 4], [c, i], [i, j, 5])$ 。
- (9) 状態が q であることを確認するルール $([i, j, 5], [q], [i+D])$ 。

i 番目のセルに状態 p から状態 q に移る j 番目の状態遷移である $(p, (ab, 1, cd), q)$ に対応する SPG の頂点とルールを図3で示す。

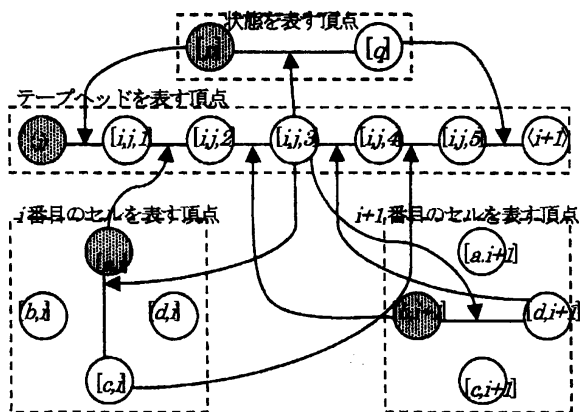


図3 $(p, (ab, 1, cd), q)$ の頂点とルール

図中塗りつぶした頂点は石の置いてある頂点を表し、STM の状態が p 、テープヘッドは i 番目のセル上にあり、 i 番目と $i+1$ 番目のセルにはそれぞれ記号 a , b が書かれていることを表している。頂点 (i) から頂点 $(i+1)$ に、または頂点 $(i+1)$ から頂点 (i) に石が移動を完了した場合、 M で状態遷移 $(p, (ab, 1, cd), q)$ が実行されたことに対応する。移動が完了する前に途中で引き換えした場合は移動する前の状態に戻る。また $(p, (ab, 0, cd), q)$ のときも同様に構成できる。

このような SPG の構成で、STM の入力受理問題が SPG 問題に多項式時間還元可能であることが示せる。SPG の構成は多項式時間でできるので、SPG 問題は PSPACE 完全であることが示された。(証明終)

4. SPG の変形

SPG 問題を GBP 問題にそのまま還元しようとするとうわりにくいので、還元しやすいよう SPG を以下のように2段階に分けて変形する。

- i. 無向辺で連結された頂点上には唯一つの石しか存在しないようにする。
- ii. 無向辺同士が交差しないようにする。

(i. の変形)

無向辺で連結された頂点上には唯一の石しか存在しないように SPG $G = (X, R, S, T)$ を変形し、新たに $G' = (X', R', S', T')$ を構成する。

$X' = \{[x, i] : x \in X, 1 \leq i \leq |S|\}$ とする。ただし $|S|$ は S の要素数である。 G において i 番目の石が頂点 x にあるとき G' において頂点 $[x, i]$ に石を置く。

R' には $(x, y, z) \in R, y, z \in X$ に対応した $([x, i], [y, j], [z, i])$, $1 \leq i \leq |S|$, $1 \leq j \leq |S|$ のルールを含める。

$S' = \{[s_i, i] : s_i \in S\}$ とする。つまり最初にある石の数は G と変わらない。

$T' = \{[t, i] : t \in T, 1 \leq i \leq |S|\}$ とする。 $[t, i]$ のいずれかに石が来たとき G' は解ける。

次に G' から $G'' = (X'', R'', S'', T'')$ に変形する。

G' がルール $([x, i], [y, j], [z, i])$, $1 \leq i \leq |S|$, $1 \leq j \leq |S|$ を含むとき、 G'' の頂点 x_i, y_j, z_i およびルールを図4のように構成する。ここで x_i は $[x, i]$ に対応する頂点、 y_j は $[y, j]$ に対応する頂点、 z_i は $[z, i]$ に相当する頂点とし、 n は S' の要素数、 m は R' の要素数とする。また記号 V, E は頂点要素の区別をつけるための便宜的な記号である。

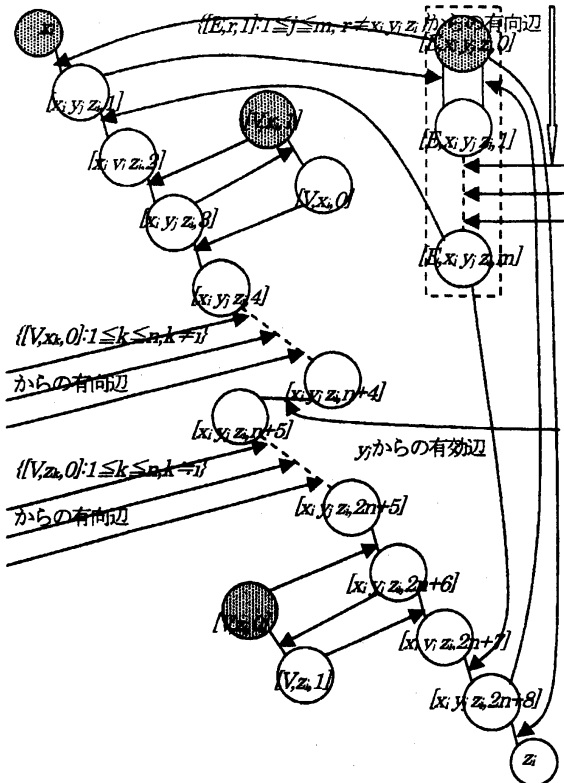


図4 G' がルール $([x, i], [y, j], [z, i])$ を含むときの G'' における頂点とルール

頂点 $[V, x_i, 0]$ および $[V, x_i, 1]$ は x_i に石があるかどうかを表す頂点であり、 x_i に石がない場合 $[V, x_i, 0]$ に石が置かれ、 x_i に石がある場合は $[V, x_i, 1]$ に石が置かれる。これらは X' の各頂点に対して定まる。

頂点 $[E, x_i, y_j, z_i, k]$, $1 \leq k \leq m$ は R' の各ルールに対して定まる。最初は $[E, x_i, y_j, z_i, 0]$ に石が置かれ、ルール $([x, i], [y, j], [z, i])$ を適用して x_i から z_i へ、あるいは z_i から x_i へ石を移動する途中は $[E, x_i, y_j, z_i, m]$ に石が置かれる。 $[E, x_i, y_j, z_i, 0]$ から $[E, x_i, y_j, z_i, 1]$ へ石を移動するためには図4の $[x_i, y_j, z_i, 1]$ または $[x_i, y_j, z_i, 2n+8]$ に石が置かれている必要がある。また $[E, x_i, y_j, z_i, 0]$ から $[E, x_i, y_j, z_i, 1]$ へ石を移動させるためには図5の各 $[E, r_i, 0]$, $2 \leq i \leq m$ に石が置かれている必要がある。このような仕組みでは頂点 x_i から z_i に石を動かしている途中で他のルールで石を動かさない。また $S'' = \{[V, s, 1] : s \in S'\} \cup \{[V, t, 0] : x \in S', t \in X'\} \cup \{[E, r, 0] : r \in R'\}$, $T'' = T'$ となる。このような変形をした G'' は G を模倣することができる。 G から G'' への変形および G' から G'' への変形は多項式時間でできる。

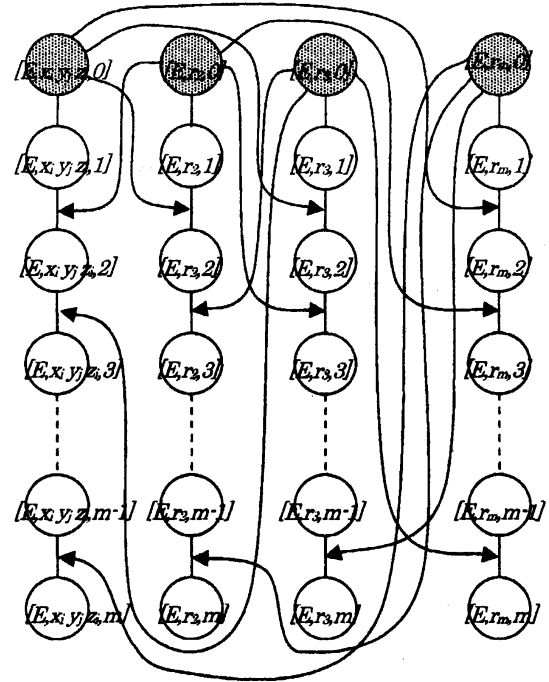


図5 頂点 $[E, r_i, k]$

(ii. の変形)

G'' の無向辺の交差がないように変形して $G'' = (X'', R'', S'', T'')$ を構成する。図6のような無向辺 uw と xz が交差する場合は図7のように変形する。ここで $(u, v, w), (x, y, z) \in R''$, $u, v, w, x, y, z \in X''$ とする。

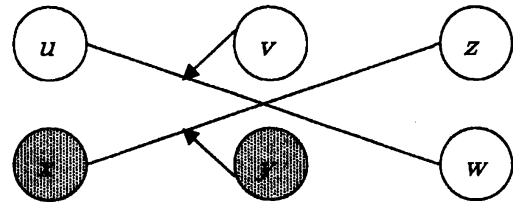


図6 無向辺 uw と xz の交差

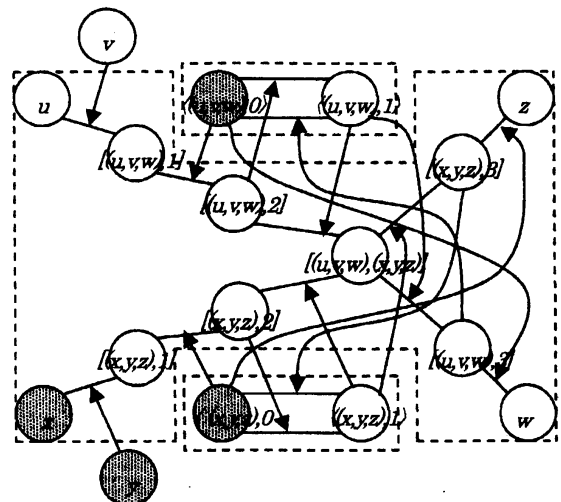


図7 図6を変形した頂点とルール

G'' の構成上、頂点 u, v, w, x, y, z の間では石は高々ひとつしか存在しないので頂点 $u-v$ 間と $x-z$ 間と移動でお互いに干渉が起きないことに注意する。

このような構成で SPG を平面グラフにすることができる。 G'' から G^p への変形も多項式時間でできる。

5. GBP 問題の PSPACE 完全性

定理 2. GBP 問題は PSPACE 完全である。

(証明) テープに GBP の盤面を記憶させるようにし、次に動かせる盤面をテープに書くように非決定性 TM を構成すれば GBP が PSPACE で解けることが言える。

GBP 問題が PSPACE 困難であることを示すには定理 1 の証明中の変形した G^p に対し

SPG が解ける \Leftrightarrow GBP が解ける

を満たす GBP を多項式時間で構成すればよい。

頂点を 3×3 の大きさのマスのマスで対応し、石を 3×3 の大きさの駒のない空白部分で模倣する。無向辺を 3×3 の駒を並べて対応し、有向辺を 1×6 または 6×1 マスの駒を並べて対応する。ルール $(x, y, z) \in R^p$ を実行するときは無向辺 xz 間を、頂点 x に対応するマスから頂点 z に相当するマスに向かって空白部分を動かしていく。ここで途中に無向辺 xz に対応する駒上を塞ぐように有向辺 y_x に対応する駒が配置されている。もし頂点 y に石がある場合は、頂点 y に対応するマスに空白部分があるので有向辺 y_x に対応する駒、頂点 y に対応するマスの空白に移動することによって空白部分を頂点 z に対応するマスに到達させることができる。また G^p の T^p に含まれる頂点に空白部分が到達したときに、「娘」に相当する駒が盤の外に出られるように構成する。以下構成の各部分の図を示す。なお図 8 から図 16 の通路になっている部分以外の薄い灰色で色付けした部分の駒は、通路になっている部分の濃い灰色で模様付けした駒の移動に関わらず動くことはない。また白色の部分は駒がない空白部分を表す。以下図を用いて G^p の頂点、辺などの部品を構成する。

図 8 は G^p の無向辺の構成である。図 8 の上の例では無向辺を通して石が移動するとき、無向辺に対応する 3×3 の大きさ駒は 3×3 マスの単位で方向に沿ってずらしていくことで 3×3 の大きさの空白部分が左または右に動く。下の例では左または下に空白部分が移動する。同様に上と下、左と上、右と上、右と下を結ぶ無向辺も構成できる。

図 9 は有向辺の構成である。図 9 の上の例では有向辺に対応する 1×6 の大きさの駒が 1×1 マスずつ

ずれていくことによって 1×1 マスの空白が右または左に下に移動する。下の例では左または下に空白部分が移動する。図 8 と同様に他の向きの有向辺も構成できる。

図 10 は頂点 y の構成である。 3×3 の大きさの駒をずらして頂点 y に相当する位置に空白部分が移動したとき、頂点 y に石が存在することを示す。またそのときルールの有向辺に対応する駒を、頂点 y を模倣するマスに 3×1 マス分だけ移動できる。 3×1 マス分の空白のうち 2×1 マス分は残り、 1×1 マス分のだけ下部に移動する。頂点 y から複数の有向辺が出ている場合は図 11 のように構成する。図 11 は 2 個の有向辺しか出ていないが、同様にして 3 個以上の有向辺がある頂点も構成できる。頂点 y が複数の無向辺と隣接している場合は図 12 のように構成する。図 12 は 3 個の無向辺と隣接しているが、同様にして 4 個以上の無向辺がある頂点も構成できる。

図 13 は無向辺 xz と有向辺 y_x が接する部分の構成である。図 10 で頂点 y に対応するマスに空白が存在する場合、有向辺に対応する駒をずらすことにより頂点 y から 1×1 マスの空白部分が上から移動する。このとき図 13 上の 2×1 の大きさの空白部分とあわせて 3×1 の大きさの空白部分ができる。その空白部分へ無向辺 xz 上を塞いでいる有向辺に対応する駒を移動することにより無向辺に対応する駒が左右にずらすことができ、 3×3 の大きさの空白部分は辺 x に相当するマスから辺 y に相当するマス向かって移動することができる。

図 14 は無向辺と有向辺が交差する場合の構成である。図中有向辺の上部または下部から 1×1 の大きさの空白部分が移動する場合は有向辺の下部または上部に 1×1 の大きさの空白部分を移動することが出来る。また無向辺の左または右から 3×3 の大きさの空白部分が来る場合は無向辺の右または左に 3×3 の大きさの空白部分を移動することが出来る。それ以外の空白部分の移動はできない。

図 15 は有向辺同士が交差する場合の構成である。有向辺の上部または下部から 1×1 の大きさの空白部分が来る場合は有向辺の下部または上部に 1×1 の大きさの空白部分を移動することが出来る。また無向辺の左または右から 1×1 の大きさの空白部分が来る場合は有向辺の右または左に 1×1 の大きさの空白部分を移動することが出来る。それ以外の空白部分の移動はできない。なお G^p の構成上、無向辺同士が交差することはない。

図 16 は出口の構成である。黒い部分は GBP の外壁に相当する部分である。頂点 $t \in T^p$ に対応する位置に空白部分が来た場合「娘」の駒が出口に到達できる。図 16 では頂点 $t \in T^p$ しか示されていないが、同様に T^p に含まれる頂点が 2 個以上の場合も構成で

きる。

これらの図の構成で G^p のすべての部分が GBP で構成できる。SPG 問題は多項式時間で GBP 問題に還元できるので GBP 問題が PSPACE 完全であることが示された。(証明終)

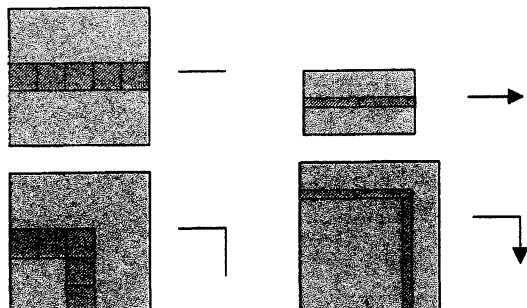


図 8 無向辺の構成

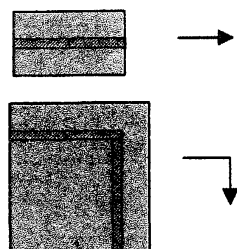


図 9 有向辺の構成

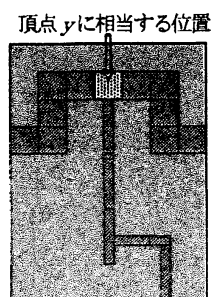


図 10 頂点 y の構成

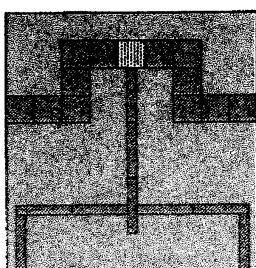


図 11 複数の有向辺が出ている頂点の構成

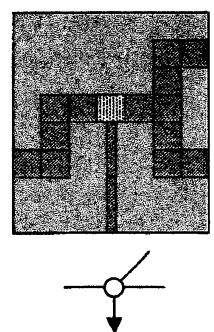


図 12 複数の無向辺と隣接している頂点の構成

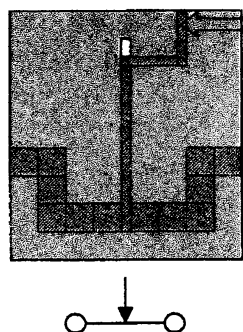


図 13 無向辺 xz とルール $(x, y, z) \in R^p$ の構成

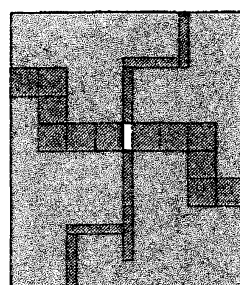


図 14 無向辺と有向辺が交差する場合の構成

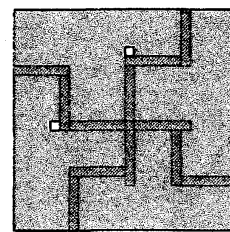


図 15 有向辺同士が交差する場合の構成

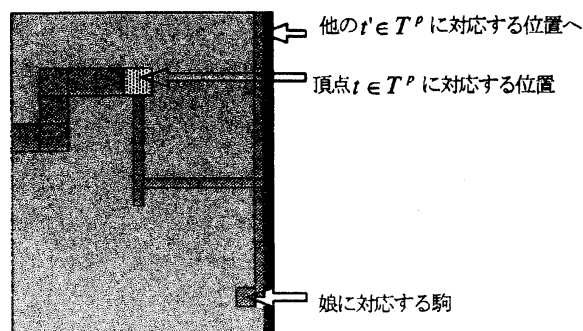


図 16 T^p の構成

参考文献

- [1] Martin Gardner, Mathematical games, *Sci. Amer.* **210**, 122-130 (1964)
- [2] Robert A. Hearn and Erik D. Demaine, The nondeterministic constraint logic model of computation: reduction and applications, *Lecture Notes in Computer Science* **2380**, 401-413 (2002)
- [3] Richard Kaye, Minesweeper is NP-complete, *Mathematical Intelligencer*, **22**, 9-15, (2000)
- [4] Joseph Cullberson, Sokoban is PSPACE-complete, *In Proceedings of the International Conference on Fun with Algorithms*, 65-76 (1998)
- [5] 横田雅也, 築地立家, 北川智博, 諸橋玄武, 岩田茂樹, 一般化詰将棋問題の指数時間完全性について, *信学論, D-I*, J84, 239-246 (2001)
- [6] Takumi Kasai, Akeo Adachi and Shigeki Iwata, Classes of pebble games and complete problems, *SIAM J. Comput* **8**, 574-586 (1979)
- [7] Harry R. Lewis and Christos H. Papadimitriou, Symmetric SPACE-bounded computation, *Theoretical Computer Science* **19**, 161-187 (1982)